

COPY
BOX
1/2

Hello Brewster & Harry!

Hope all is well with you. The telecommuting
is working out pretty well, and I'm having
fun in Minnesota.

Tom

A Preliminary Design for a 3-D Spatial User Interface for Internet Gopher

Mark P. McCahill
Distributed Computing Systems & Services
University of Minnesota

Thomas Erickson
User Experience Architect's Office
Apple Computer

Introduction

This paper describes the beginnings of an exploratory design project: its aim is to sketch out a starting point for the design of a 3-D spatial interface to a large, internet-based information system, and to capture some of the initial design rationale. After this, we intend to refine the design, implement it, make it freely available over the internet, and observe the reactions to and usage of the interface.

Although we will discuss the rationale for a 3-D spatial interface in more depth in the paper, it is worth summarizing our basic motivations up front. First, we hope to learn about the design issues that are raised by 3-D spatial interfaces. Although 3-D interfaces have been getting increasing attention from researchers, little attention has been devoted to the design tradeoffs involved. Second, most of the 3-D spatial interfaces in existence — particularly those that provide interfaces to information spaces — are in research laboratories; the opportunity to create an interface that will provide an extremely broad range of people with access to a public information space of millions of items seems too good to pass up, and with the advent of RISC-based personal computers, the time seems ripe. Finally, we would like to put our beliefs about the advantages of 3-D spatial interfaces to the test. We see two possible advantages: First, 3-D seems to provide the possibility for representing many dimensions of information and meta-information in a compact and natural way. Second, we believe that in the long run information spaces will also be used as social spaces, and that 3-D can serve as a natural framework for supporting social interaction (Erickson, 1993).

In this paper we begin by describing internet Gopher as it exists today. Then we discuss some of the known problems that we hope to address, as well as some new prospects opened by moving to a new type of interface metaphor. We distill these problems and prospects into design criteria, and then sketch out the basic design with comments on the design rationale inserted as appropriate.

Internet Gopher Today

Internet Gopher is an information system used to publish, organize, and find information distributed across the Internet. Gopher consists of information servers which may reside anywhere on the internet, and a number of clients which are used to access information on the servers. Gopher is a decentralized system: Gopher servers are administered by a wide range of individuals and organizations — virtually anyone who has access to information and to a machine on the internet can, with a little technical know-how, create and maintain an information server. Similarly, Gopher client applications are widely available and provide access to any of the

information servers on the internet, with the exception of a small number of limited-access servers. To the individual user Gopher appears to be exceptionally simple: it provides access to a hierarchy of information which the user can browse; the fact that information is stored at different places on the internet is typically invisible to the user. We will refer to the sum of all Gopher-accessible information on the internet as Gopherspace.

Since its introduction in June, 1991 Gopher has become quite popular. In December 1993, there were over 4800 gopher servers accessible on the Internet, and well over 1.5 million items accessible in Gopherspace. By March 1994 there were 6700 Gopher servers accessible over the Internet. Gopher traffic across the NSFnet backbone has been increasing at an average rate of 20% a month during the last year, and Gopher's share of the total NSFnet traffic has increased to about 3.5%. There are now at least 4 different Macintosh Gopher clients, 5 Windows clients, 4 clients for Unix/X-windows, 2 Amiga clients, a VMS client, and even Gopher clients for MVS and VM/CMS. All these clients have conceptually similar user interfaces.

The typical gopher client presents users with a directory hierarchy to navigate (figure 1). Each gopher directory may contain items of four types: documents, search engines, icons for launching telnet sessions, and other directories. Gopher clients typically display the contents of a directory in a window, with an icon representing the type of item followed by the name of item; the name of the directory is used as the title of the window. Some clients restrict the user to viewing each successive directory in a single window, while other clients allow for multiple windows (each successive directory is displayed in a new window). Gopher clients provide no representation of a Gopher Server as a whole: an item in a directory of one Gopher server may be a link to an item on another Gopher server so that, to the user, boundaries between servers are invisible.

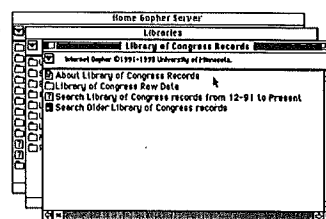


Figure 1. A typical client, portraying gopherspace as a hierarchy of folders.

Motivation

The design is motivated by a mix of pragmatic and exploratory impulses. On the one hand, there are several usage problems that would be difficult to solve within the constraints of the current interface metaphor. And, on the other hand, the advent of RISC-based personal computers means that there will be sufficient power to support a whole range of new behaviors, offering the prospect of transforming information systems into more flexible, information-rich environments. We will begin with a brief discussion of the motivating factors, and will then summarize these as design criteria.

Problems and Prospects

While the current user interface is popular, there are three well known usage problems.

The Lost-in-Space Problem

Users complain of feeling lost after navigating for a while and have difficulty remembering where they found an interesting item. In part, this is due to the absence of any global representation of the structure of information hierarchy, and in part because the path followed by a user is either invisible or, at best, implicitly embodied in a stack of directory windows. The 'lost-in-space'

problem has been observed in a number of other domains, most notably hypermedia. Users need an overview of gopherspace, within which they can see their locations and the paths they have followed.

The Grouping Problem

Within a directory it is difficult to show relationships between items represented in a linear list. Some server administrators resort to putting items with blank names in their directories to group clusters of items. An analog of this problem occurs in lists of results generated by search engines. The results are typically sorted by relevance (as ranked by the search engine), but the user interface doesn't have a good way to convey their relative relevance. And, as in directories, it is difficult to show the clustering of related sets of documents. Ideally, both relevance to the search terms and "closeness" to other documents (along a variety of user-specifiable dimensions) ought to be visible to the user at a glance.

The Browsing Problem

It is difficult to browse because documents reflect so little of their content. All that is available is the item's name and the information about the document's type embodied in the icon. The user's only other option is to open the document—often a time consuming process—and see everything in the document. Neither option is supportive of browsing: users need to see more information about the content of a document without there being so much that they are unable to compare and contrast different documents.

In addition to remedying today's usage problems, there are a number of intriguing prospects which a new interface could open to exploration:

Interaction Traces

Users browsing a large information space could benefit from the activities of previous visitors. For example, users are often interested in knowing about the relative popularity of documents, as judged by the frequency with which they are viewed or copied. The idea behind interaction traces is that this could be reflected in the visual appearance of the document's icon. A number of researchers have explored ways in which visible representations of computational objects can reflect traces of the ways in which users have interacted with them (e.g., Hill, et al., 1991; Wroblewski, et al., 1994).

Providing a Sense of Place by Customization

Today, gopherspace is generic: any gopher directory looks like all the others, regardless of where it is or what it contains. Interaction traces would provide some differentiation, but what we'd really like is to make it possible for an area of gopherspace to reflect something of its contents, and something about those who construct it, maintain it, and frequent it. Sense of place could be initially supported by allowing server administrators control over visual characteristics of items on their servers, and allowing users to customize the appearance of individual items and directories by adding notes or graffiti.

Providing a sense of place would benefit both administrators and users. Server administrators would like to be able to customize their areas, but their opportunities for doing this within the constraints of a list of names and icons are quite limited. Supporting customization by administrators is important because many Gopher servers are maintained by volunteers with little or no institutional support or recognition; anything which can increase the gratification administrators receive from their efforts will ultimately benefit gopherspace as a whole. Users too would benefit from being able to customize items in gopherspace: it would transform Gopherspace from something 'owned' by server administrators to a more public, collectively

shaped sort of space. More generally, as areas with gopherspace develop their own sense of place, it seems likely that users would begin to recognize places and regions; and, while users may still get lost, they may begin to develop a sense for where they're lost.

Transforming Information Spaces into Social Spaces

Very often, the preferred method for obtaining information is to ask someone else; turning to an information source — either physical or electronic — is often a last resort. In fact, sometimes people search for documents only as pointers to their authors to whom they then direct their queries (Erickson & Salomon, 1991). It is also true that much information access is part of a larger, often collaborative task: people seek information because they are trying to solve a problem, test a theory, understand a concept, or communicate their understanding to others. All of this suggests that there is little reason to segregate information access from other activities.

The increasing popularity of MUDs and MUSEs for supporting conversation, teaching, and other group activities demonstrates that virtual spaces — even those that are purely textual — can serve as frameworks for a broad range of collaborative activities (see Curtis, 1992; Bruckman, 1992; Bruckman & Resnick, 1993; Erickson, 1993; Rose, 1994). Indeed, some MUDs are beginning to provide access to Internet Gopher from within their environs (e.g., Masinter & Ostrom, 1993). We propose to pursue the same end from a different direction, and to explore expanding gopherspace into a social space that can support a broad range of activities that complement information browsing and access.

Initial Design Criteria

The problems and prospects we have just covered map into a few general design criteria.

Richer Representations

There is a need for richer representations for servers, directories, and documents. The lost-in-space problem suggests the need for a high level overview of gopherspace. The grouping problem indicates the need for a representation of collections of documents that can reflect their similarities and differences along a variety of dimensions. Similarly, richer representations for individual documents would alleviate the browsing problem. Finally, the expansion of gopherspace into a social space requires representations that can provide a medium within which human-human interaction can occur.

Dynamic Representations.

Representations need to be able to change over time. Sense of place requires representations that can be customized by administrators and end users, and interaction traces require representations able to reflect the interaction history of individual documents and collections of documents.

Metainformation.

All of these changes to the representations in Gopherspace require that meta-information about servers, directories, and documents be made available via the Gopher protocol. The prospects of interaction traces, customization, and social spaces also involve meta-information, but meta-information that is external to gopherspace: it is applied by users, or inferred by the system based on user interaction.

Backward Compatibility

There is a final, very important criterion not suggested by the preceding discussion. It is important to recognize that there is a large installed base of Gopher servers and clients, and a community of

administrators and end users—Gopher is as much a social phenomenon as a technology. Backward compatibility of any new client is essential for acceptance. It will be impossible to change all of gopherspace overnight, so any new client must handle both servers that have stored additional information (e.g., about relative clustering of items in document collections) and ancient unmodified servers. This must be done without stepping outside the new user interface metaphor.

Why a 3-D spatial interface?

First, note that 3-D space as a metaphor for information is nothing new; it is deeply embedded in our language (e.g., Lakoff & Johnson, 1980). Without thinking about it, people use spatial and object-centered terms for discussing abstract concepts. We speak of getting *overviews*, seeing things from different *perspectives*, and *grasping* new ideas. Providing a 3-D spatial interface is, in part, just providing a concrete embodiment of language we already use.

Furthermore, a 3-D space is a very powerful representation system; it provides a flexible conceptual framework with which many variables can be integrated. Relationships between items can be shown by spatial proximity. Interaction traces can be represented as physical wear on objects in the space. Sound can be used to indicate activity going on within the space, but outside the view of the user. 3-D space can provide a framework within which users can interact with one another. 3-D representations also implicitly include two representations: a surround view when you are "in" the 3-D space that is suitable, for example, for viewing collections of documents; and a bird's eye view that provides an overview of the structure of the space. The fact that the same conceptual model provides two different representations that are connected in a well understood way (and that permit the transition from one to another to be shown) is very useful.

Another advantage of 3-D space is that it is familiar. People understand a lot about space. For example, they know that any 3-D object has other sides that aren't immediately visible, and they have expectations about what to do to get a view of those other sides. They are familiar with manipulating and interacting with objects in 3-D space. People are familiar with navigating through space (since people have little experience flying, we intend to implement constrained navigation, so that the user experience is something like driving a car). Besides, it'll be loads of fun..

When we are ready to expand gopherspace into a social space, 3-D spaces provide a natural way of supporting interaction between people. As human beings, we understand a lot about the social conventions of spaces. We understand the distinction between public and private spaces; we know that you have to pay to enter some spaces at which time you gain temporary rights to that space. We understand that particular activities, people, rituals, and behaviors are associated with particular types of spaces. We have spent our lives learning to recognize spatial cues: what entrances look like, what a bulletin board is, where you are likely to find a you-are-here map, and so on. All this knowledge can be leveraged in a 3-D spatial metaphor.

Finally, there is a rich vein of knowledge about how the design of 3-D spaces that is ready to be tapped. Disciplines such as architecture, landscape design, and urban design have a sophisticated understanding of the issues that arise in spatial design (e.g., Lynch, 1976; Hough, 1990; Alexander, 1987; Southworth, 1969) that may be applied to the design of virtual environments. In addition, many researchers have studied ways in which spatial environments support (or inhibit) human-human interaction (e.g., Whyte, 1988). Finally, it is interesting to note that urban design, in particular, has striking parallels with the situation faced in designing large, distributed information spaces: in both cases the designer must create a framework that is sufficiently robust

that third parties can come along and add unforeseen things without destroying the coherence and consistency of the design.

The Preliminary Design

In this section we sketch out the preliminary design for the interface, with some comments on the rationale for various decisions. It is important to emphasize that this is the starting point, not the ending point. In general, the preliminary design is based on a combination of analysis and intuition; at this point, no testing or prototyping has been done, with the exception of a few mock-ups of 3-D icons and neighborhoods generated to facilitate discussion of how to design legible 3-D icons, and how to support navigation among them. We take it as a certainty that as we proceed both implementation constraints and feedback from prospective users will shape the design in major and unforeseen ways. We expect the implementation of the design will proceed through (at least) two stages: first, we will focus on creating a 3-D information space; only later will we integrate the functionality necessary for supporting social interaction.

For expository purposes we will describe the preliminary design in terms of three levels of representation — the overview, the neighborhood, and the individual item. We will begin with the neighborhood, the representation for the collection of items with which the user is interacting. Next we will examine some details of the 3-D icons used to represent individual items. Then we'll look at the representations which provide the overviews of regions and of gopherspace as a whole. Finally we will describe interactions in gopherspace, including: opening documents, navigating around a neighborhood, moving from one neighborhood to another, and user-user interaction.

The Neighborhood

The neighborhood is the representation of the collection of items with which the user is interacting. Neighborhoods are either constructed by server administrators (as a directory in the the server's file hierarchy) or generated by search engines in response to a user-entered query.

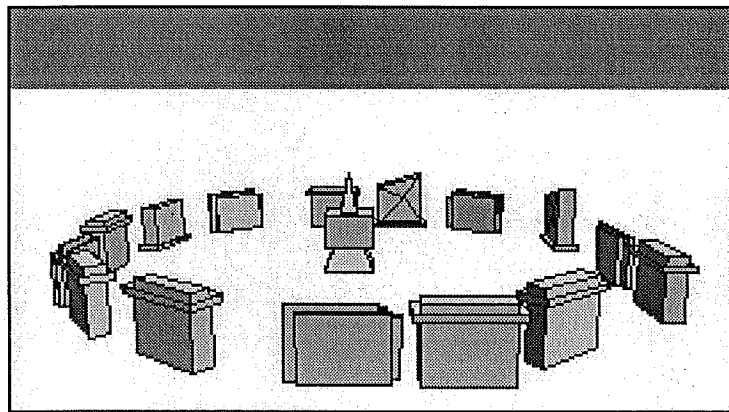


Figure 2. The circular "Stonehenge" icon arrangement.

The Arrangement of the Icons

We explored two representations of neighborhoods: circles and 'streets'. Circular arrangements of items have a number of strengths. First, the user will generally have a straight on view of the fronts of a couple of 3-D icons, which is valuable because the fronts of these icons will generally contain details of their content. Since the user enters the neighborhood near the center of the circle

of icons, the user is always going to be looking at something and it is difficult to drive off into limbo. A second useful property of a circular arrangement is that it is easy for the user to understand: the user can quickly get an idea of how many icons are in the neighborhood (based on the density of icons and the radius of the circle). The circular arrangement of icons defines an enclosed space that may be used as a collective gathering space for users. The circular arrangement also defines a center point, at which we will place a 3-D 'kiosk' icon that will function as the user's link back to the previous neighborhood, and as a sort of information desk for the neighborhood. If we allow for display of user-entered comments from the people who have visited this directory this should also appear on the kiosk.

The street metaphor was investigated and rejected because the user is either facing down the axis of the street and has an oblique view of most of the faces of the icons, or is facing one side of the street and is required to turn fully around to view the next closest icon. It also may be difficult for users to tell how long a street is, and unless the street is short, it really has no center or natural gathering point. Finally, simple mock-ups of streets in a 3-D modeling program resulted in arrangements that felt very claustrophobic, since fairly large 3-D icons were necessary for information display purposes.

A variant of the circular arrangement of items is the spiral. We intend to use the spiral arrangement for collections of icons generated by search engines in response to user queries. Formally, the spiral has a nice family resemblance to the circular arrangement so that it too defines an enclosed area with a center point; at the same time, its greater openness and dynamicism seems a good reflection of the transitory nature of most queries. Also, a spiral has directionality, and thus provides a natural ordering within which the relevance of items to the query can be reflected. That is, the more relevant the items, the closer they are to the root of the query; and, more generally, a search that returns a large number of very relevant items will have a tightly coiled spiral, whereas one with few relevant items will have a very loose spiral.

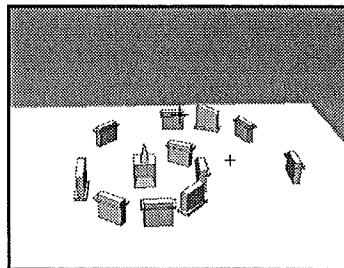


Figure 3. Results of a search arranged in spiral fashion.

Sound

We intend to support the use of sound as part of a representation for a neighborhood. Sound is valuable because it can maintain the sense of being in a particular place, even when the place is too big or complex to be shown all at once. Server administrators should be able to define a digitized sound for sound-savvy clients to play while the user is within the directory... hopefully unobtrusive sounds similar to some of Brian Eno's ambient music.

Sound can play a variety of other roles. It may be used to reflect activity of other users in the same or nearby neighborhoods (Gaver, et al, 1991; Cohen, 1993). Variations in its timbre could be used to give hints as to the size of the neighborhood (Gaver, 1989). In the physical world, sounds also play an important role in supporting the feeling of a sense of place (Southworth, 1969).

To make sound a common part of all 3-D space will require synthesizing sounds for gopher servers that do not provide a server-defined sound. Having the client software generate an ambient wind sound attenuated by the number (and type) of objects in the scene is probably the best approach creating sounds for servers that do not have their own. By making the attenuation of the ambient wind sound depend on the objects in the local space we can get automatically create variation in tone and timbre.

Interaction Traces

If usage information is available from the server, footprints (or some sort of dirt on the ground) can be used to show which of the items in the neighborhood are the most popular. This is like the worn marks on subway platforms in New York. You can predict where the doors of the subway train will be when it stops by looking at the worn spots on the platform. The same sort of cue can be used in a neighborhood to show users who want to follow the beaten path, where that path lies.

The 3-D Icons

In our design, 3-D icons can vary along three dimensions: form, surface characteristics (color and texture), and information about the icon's content (name and proxy).

The Forms of 3-D Icons

The basic form of a 3-D icon is an approximately rectangular box that represents the type of the object. Box-like icons are attractive since they keep the scene rendering requirements to a minimum by keeping the number of polygons down and simplifying hidden surface removal. Box-like objects also provide plenty of space to map textures, draw items names and other information, and can look vaguely like buildings that people encounter in life.

The form of the icon indicates the type of object it represents: the constraints on the icons' forms are that the icon's type should be recognizable from any direction (and ideally from a distance), that most icons have a large, flat surface area on which its proxy may be displayed, and that the form be relatively simple so that large directories displayed by clients on low-end machines not take excessively long to render. Figures 4 through 9 show initial passes on the forms for types of icons.

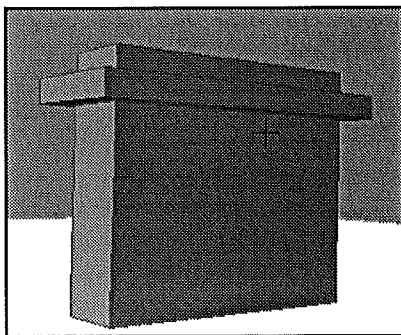


Figure 4. Document icon.

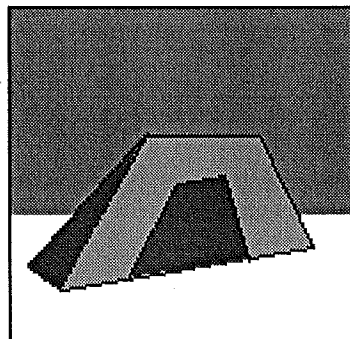


Figure 5. Gateway icon

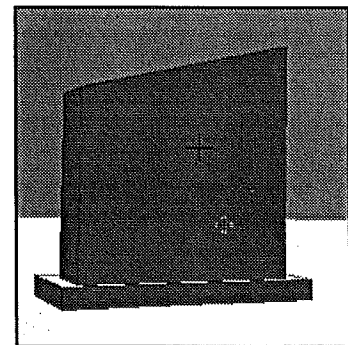


Figure 6. Search engine icon

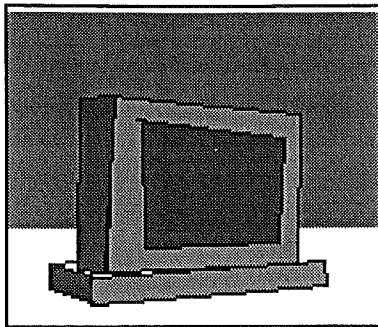


Figure 7. Interactive session icon

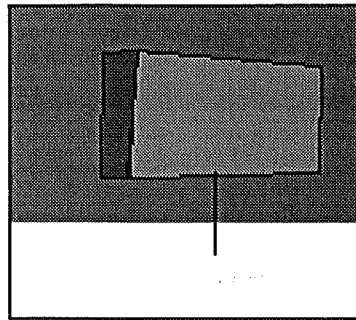


Figure 8. Person icon

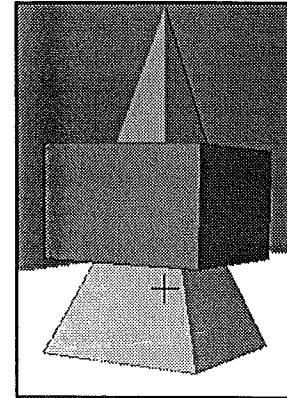


Figure 9. Kiosk icon

The objects represented by the icons are mostly analogous to types of icons in gopherspace today (documents, search engines, interactive sessions, and gateways to other directories). Two new types are kiosk and person icons: kiosks have been described in the preceding section; person icons are place holders for icons which will represent users when the environment is able to support synchronous social interaction.

Surface Characteristics of Icons

Surface characteristics of icons include color and texture. At the moment our intent is to use color as redundant information, to indicate the type of the icon. The advantage of this is that it will allow types to be distinguished in overviews. Texture is a variable the server administrators will be able to customize; they will be able to define the texture as a bitmap that will be painted onto the surface of the icon, and onto which other information (e.g. proxies) will be mapped).

Information about the Content of the Icon

The face of the 3-D icon is divided into areas used for the name of the item and the proxy. The title of the item is written across the top: on document icons there is a ridge wrapped around the icon about 80% of the way up the icon where the title is written; other icons have the title in the same location but without the ridge. Below the title is where the proxy is displayed. The proxy reflects something of the content of the object represented for the icon: for a picture, it might contain a thumb-nail sketch; for a text document it might contain key words; for a new neighborhood, it might contain an indication of the neighborhood's size. Researchers have suggested a number of possibilities for proxies which would be interesting to explore in this context (Houde, et al. 1993; Wroblewski, et al., 1994).

As the user approaches a 3-D icon, the amount of information displayed on the icon changes since there is more screen real estate to use for display and since the user is presumably more interested in the item. From a long distance, only the general outline and color of the icon is readily discernible. At middle distance the texture map and name of the icon are visible. At close range proxies for the information within the icon become visible; as the user approaches even closer to an icon, more information might become available.

What the proxies are will vary depending on the type of object. Directory icons may show a rough rendering of the content of the directory (i.e. the number and arrangement of the icons contained in that directory). Document proxies are probably the first part of the document or a diskette icon to show that the contents is a binary file. The document proxy referring to a Quicktime video might contain the poster view of the video. Sounds could also be used as part of the proxy of an item, perhaps brought into play when a user comes very near the icon: a directory's proxy might use

sound to give a hint of what the new neighborhood is like before the user enters it; a document's proxy might use whispered text-to-speech to provide more detail about a document's content; and of course icons for audio and video files could just play part of the soundtrack.

Representing Overviews of Gopherspace

The purpose of an overview is to give users a glimpse of the larger context in which they are situated. In the current design, all overviews take the form of 2-D maps, which can be brought up in a window separate from that containing the 3-D view. There are three sorts of overviews that seem likely to be of use: an overview of the neighborhood; an overview of the local region of gopherspace; and an overview of all of gopherspace.

Overview of the Neighborhood

The collection of items will either be the current directory or the results of the most recent search. The overviews will 2-D projections of the neighborhood as seen from overhead: that is, either the "stonehenge" arrangement or a spiral of icons. Colors will enable users to distinguish the type of the icon (since, in a 2-D projection this will probably not be readable from the icon's form). The specific ways in which these overviews will be used is yet to be determined, but such views could obviously provide the user with a you-are-here orientation, and a way of quickly assessing the size and relative proportion of icon types in a particular collection. A miniature version of this overview could also be used as a proxy for a gateway icon.

Overview of the Region

We haven't decided precisely what a region should be. We will take as our working definition that a region is comprised of the current neighborhood, neighborhoods directly connected to that neighborhood by gateways, and (perhaps) neighborhoods connected to those. An alternate possibility is to define the region as the server the user is currently on. Note that since any of the gateways may be to directories on different servers, the two definitions of a region are quite distinct, rather than the second being a superset of the first. One consequence of this is that the first type of neighborhood will have to be computed on the fly by querying other gopher servers for the structures of relevant (i.e. connected) neighborhoods. Given a definition of a region, the region overview will be generated in the same way as that for the individual neighborhood (see figure 10 for an abbreviated example).

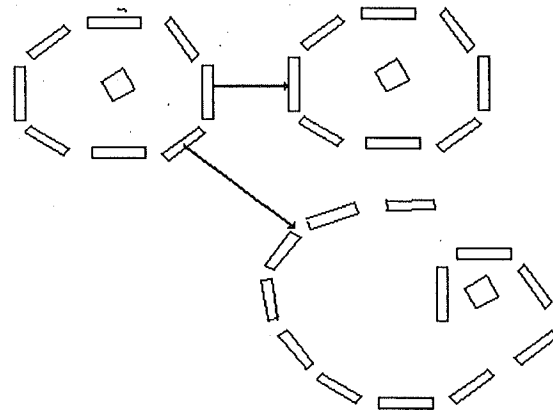


Figure 10. Overview of a (small) region: a query neighborhood, with two connected directory neighborhoods.

Overview of Gopherspace

Finally, an overview of all (or at least large portions) of gopherspace would be useful. However there are two practical problems. First, it is difficult to identify the contents of gopherspace. Because Gopher is a distributed system without a compulsory server registration, there is no global list of all servers in gopherspace. Maintaining such a list is a daunting prospect given the growth in number of gopher servers (3400 in September 1993, 4800 in December 1993, 6700 in March 1994). The second problem is that the contents of gopherspace are constantly changing:

new servers come on line, other servers go off line, directories are added, deleted or reordered, and items are added or deleted. The most feasible alternative is to encourage individuals within the Gopher community to use software to generate maps, which could be made available on map servers for those who wish them. Although it may strike the reader as a case of wishful thinking, similar efforts have produced gopher-related tools like Veronica and Archie.

Assuming the existence of a relatively up-to-date list of the contents of gopherspace, the question of how to represent the overview of gopherspace remains to be dealt with. Given the magnitude of Gopherspace—thousands of servers, millions of items—representing it (or even significant subsets of it) with miniature 2-D projects of the neighborhood is out of the question. Even were this possible, representing a dynamically growing space using a Euclidian plane is problematic: when new servers, neighborhoods, or items are added, the new items must be put somewhere, and this will often result in distorting the structure of the overview. This is a problem insofar as one of the purposes of an overview is to provide a consistent frame of reference for the user.

A solution that addresses both of these problems is to use a more symbolic overview, projected on a frame of reference that is independent of the content of gopherspace. The obvious candidate is a map of the physical world. This has three advantages. First, people are relatively familiar with the structure of the physical world. The advantage here is simply one of mnemonics: people have a better chance of remembering where they saw something interesting if they can associate it with a known physical locale (in fact an ancient mnemonic technique known as the method of loci is based on this). The second advantage of a map of the physical world is that although Gopher servers can, in principle, be located anywhere, in fact they are often located in physical proximity to the groups that maintain them. Thus, the relationship between real world location and the type of information server found there is not wholly arbitrary. Thus, someone interested in information about visiting Japan might well want to look for Gopher servers in Tokyo; someone interested in Artificial Intelligence would be well advised to search MIT, CMU, and Stanford for servers; someone interested in Blues music might try New Orleans or the Smithsonian. The third advantage of a map-based overview is simply that we believe people will like it. It will emphasize their ability to virtually travel across the world, finding information in various locales, and will give people a sense of power and excitement.

In the absence of programmatically generated overviews of gopherspace, one initial step would be to have the client construct overviews of the areas of gopherspace that the user has visited. While this will be useful only for seeing where one has been, and not for aiding exploration of new areas of gopherspace, it is better than nothing. And this would provide placeholders in both the user interface and protocols for more comprehensive overview information that may eventually become available.

Interaction in Gopherspace

In this section, we briefly summarize the sorts of interactions that can occur in gopherspace.

Navigating within a Neighborhood

Most people are familiar with navigating 3-D spaces since this is something they do everyday of their lives to get from one place to another. On the other hand, most people have little experience flying through space, so by limiting the number of options the user has for flying into limbo, we can make the user experience similar to some of the better arcade style games (like Spectre). By limiting the user's navigational controls to forward and back turn left, turn right we can make it easy to learn how to use the interface. By limiting the height of the 3-D icons, we can ensure that

the icon's proxy will usually fit within the user's field of view, and thus we needn't worry about providing ways to look up or down, or left or right.

Interaction Details: Steering and Opening

We have not yet decided on how to carry out interaction at the most basic level. It will be desirable to open (and therefore to select items); and it will be necessary to start, stop, and change the direction of movement in Gopherspace. One possibility is to ignore selection, and to allow items to be opened by driving into them (O'Sullivan, 1994). Alternatively, we can allow users to navigate around the space, and then select items and open them. In this case, either screen controls will be needed for one or both options, or use of the command key with the mouse will be necessary to disambiguate between the steering and opening modes. Onscreen controls seem to be the better option, both in terms of ease of learning and use, and because they could support a wider range of low level operations. For example, a single click might be used to choose an option to scan the neighborhood, taking the user on a circular path around the inner periphery of a neighborhood — if clicks, and click-and-holds were used to steer, this could quickly become physically tiring).

In most cases, opening an item would result in a new window being displayed on the screen. If the item is a document, the window would of course contain the contents of the document. If the item was an interactive session or search engine, it would contain whatever prompts are generated by session host or search engine. If the item is a gateway to a neighborhood, 'opening' the gateway results in a transition to the new neighborhood. If the item is the kiosk icon in the center of a neighborhood, it returns the user to the previous neighborhood.

Transition between Neighborhoods

When the user 'opens' a gateway, visual and sonic feedback should be used to make it clear to the user they are traveling somewhere else; this is a rough equivalent of the zoomrect transition used on the Macintosh when an icon is opened. This happens when a user enters a gateway or kiosk, or executes a search on a search engine. If we handle this properly, it should give the user something to look at while the client software sets up and renders the next neighborhood the user will view.

For the neighborhood-to-neighborhood transition, the goal is to produce the experience of entering the icon, emerging into an open space, and zooming into the new neighborhood (shown in figure 12). The experience would be something like a roller-coaster ride, as the user follows a predetermined course and cannot steer. If the neighborhood is on a different information server, the first part of the 'ride' might be through an abstract grid, representing a transition between servers. The amount of time spent in flight depends on how long the software needs to fetch the information from the appropriate gopher server and render the next neighborhood. Once the next neighborhood is ready to be displayed to the user the flight over the grid ends with an approach over (and then down into) the new neighborhood. The user can get an idea of the general layout of the neighborhood during the approach and landing.

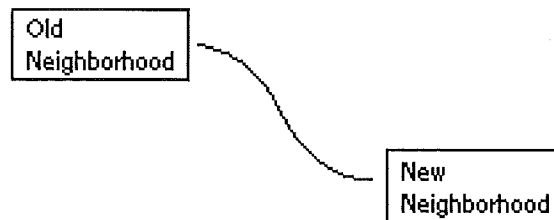


Figure 12: Trajectory between neighborhoods.

The idea behind this trajectory is to give the user a sensation of high-speed travel over an anonymous looking grid, culminating with a slow approach to the new neighborhood. It is important that the approach to the new neighborhood allows the user see the general arrangement of the neighborhood so that they can get a sense of the number and type of items in it. To make this visible, the user's trajectory is one of swooping down from above (like a plane landing). The user is deposited near the central kiosk facing so that both a part of the kiosk and some of the items in the neighborhood are visible (figure 13).

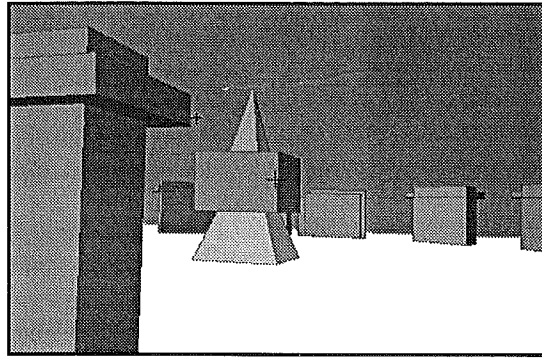


Figure 13. Initial view on entering a new neighborhood

Interacting with Other Users

The details of this remain to be worked out. However, as a very rough first pass, we'll assume that interaction occurs in a MUD-like fashion using MUD conventions, with a separate window appearing to support textual dialogs between different users in the same neighborhood (see Curtis, 1992; Bruckman & Resnik, 1993; Erickson, 1993; Rose, 1994).

System Architecture, Protocols, and Zoning

The 3-D user interface described here will be rendered and managed entirely by the client software. No changes to Gopher servers will be required, at least for the information environment phase of the implementation. Client software that can synthesize the spatial scene from current gopher directory and item meta-information allows us maintain compatibility with all of the current gopher servers. A happy side effect of this approach is that server and network bandwidth demands are minimized since we do not require servers to render scenes and ship bitmaps of the scenes over the network. Backward compatibility issues are also addressed by using the Gopher+ protocol's item attributes to hold meta-information. Gopher+ item attributes provide an open-ended, extensible way of associating arbitrary meta-information with items and directories, and methods of accepting information sent from the client, so the user interface we propose will not require a new protocol.

An exception to this client-centered approach arises when we allow server administrators to control over the appearance of servers they administer. By default, the Gopher client generates the geometry and layout of the neighborhood within the user's computer, but sense of place requires that we provide administrators with control over things like the layout within the neighborhood, sound, textures mapped onto icons, and perhaps icon geometry. We have reached no final decision on what to do, but our plan for the initial implementation is to send the client formulae for server-specific textures and sounds (as Gopher+ item attributes), which the client can then recognize. Sounds and textures (bitmaps) are easy to generate and carry a lot of information; most server administrators do not have 3-D geometry tools needed to facilitate modifying icon geometries. Eventually we would like to allow mapping Quicktime or MPEG video onto the icons

for a real Las Vegas-style user experience.

An open question is how much control to give server administrators over the look of their areas. On the one hand, server administrators could be allowed to design their own 3-D icons; they could define the icon geometry, which would then be shipped to the client for rendering. However, there are several possible problems here. Two of the most serious have to do with efficiency and consistency. For example, ornate designs might take too much processing power to render in an acceptable time on all platforms. Once server administrators can create their own icons, clients must take steps to protect themselves from badly designed icons; the clients might have to employ local zoning restrictions by refusing to render icons with excessive numbers of polygons (or non-convex polyhedra, or other items beyond the limits of the client's 3-D rendering engine). Similarly, once the server administrator has control over the layout of the icons in the neighborhood, clients will need to employ their own local zoning ordinances to check the reasonableness of the layout (disallowing overlapping polyhedra, for instance).

A more serious problem is that of consistency: users would presumably like to be able to recognize particular types of documents, regardless of what server they're on. Users entering a new neighborhood that happens to be on a different server should not have to stop and figure out what a document icon looks like in *this* neighborhood; there should be a family resemblance that users can rely on. One solution is to allow the form, or a portion of their form (e.g., the tops of the icons) of the icon to stay constant throughout gopherspace. This would allow users to assume that if the icon has a slanted top, it is a search engine. Also, the basic box shape will probably be strongly mandated since the client expects to be able to map texture, names and the document's proxy onto the surface of the icons.

Conclusions

As this is a report of work in progress, we have no real conclusions, as yet.

It is worth noting that even in these earliest stages of design, we have had to deal with issues that lie outside the realm of the disciplines traditionally involved in interface design.

- What forms should 3-D icons have so that they are both simple and yet recognizable from many directions?
- What types of spatial layouts can help people remain oriented in a large space? What sort of layouts are legible both from the inside and from the outside?
- What icon and layout geometries are most invariant over scales, allowing them to be recognized from very close, and very far away?
- How rapidly should a transition into a new spatial layout occur, so that the user can take in enough information to be oriented, but avoid boredom?
- How can we simultaneously support the sorts of regional and individual variation of appearance that makes real places rich and inviting, without sacrificing the core of consistency that that allows people to stay oriented and comfortable in real spaces?

The next step is to implement a working prototype on a PowerPC. As of this writing, a very crude prototype, complete with rendering engine, is running.

Acknowledgements

This paper is based on dreams and discussions that have been going on within part of the Gopher software development team (Paul Lindner, Farhad Anklesaria, David Johnson, Neophytos Iacovou) at the University of Minnesota over the last two years. We have also learned from the work on YORB, carried out by Dan O'Sullivan and his colleagues in the New York University School of Interactive Telecommunications.

References

Alexander, C., Neis, H., Anninou, A., & King, I. (1987) *A New Theory of Urban Design*. New York: Oxford University Press.

Bruckman, A. (1992) Identity Workshop: Emergent Social and Psychological Phenomena in Text-Based Virtual Reality. Unpublished manuscript, 1992. (Available via anonymous FTP from media-lab.mit.edu, /pub/ MediaMOO/Papers/ identity-workshop.)

Bruckman, A & Resnick, M. (1993) Virtual Professional Community: Results from the MediaMOO Project. Presented at the Third International Conference on Cyberspace, May 1993. (Available via anonymous FTP from media-lab.mit.edu:/pub/MediaMOO/Papers/ MediaMOO-3cyber-conf.)

Cohen, J. (1993) Monitoring Background Activities. *Proceedings of the First International Conference on Auditory Display*. Santa Fe, NM.

Curtis, P. (1992) Mudding: Social Phenomena in Text-Based Virtual Realities. *Proceedings of DIAC '92*. Available via anonymous FTP from parcftp.xerox.com, pub/MOO/papers /DIAC92.

Erickson, Thomas. (1993) "From Interface to Interplace: The Spatial Environment as a Medium for Interaction." *Proceedings of the European Conference on Spatial Information Theory*. Heidelberg: Springer-Verlag.

Erickson, T. & Salomon, G., (1991) Designing a Desktop Information System: Observations and Issues. *Human Factors in Computing Systems: the Proceedings of CHI '91*. ACM Press.

Gaver, W. W. (1989) The Sonic Finder: An Interface that Uses Auditory Icons. *Journal of Human-Computer Interaction*, 4:1. Lawrence Erlbaum Associates.

Gaver, W. W., O'Shea, T. & Smith, R. B. (1991) Effective Sounds in Complex Systems: The ARKola Simulation. In *Human Factors in Computing Systems: the Proceedings of CHI '91*. Austin,TX: ACM Press.

Houde ,S., Salomon ,G. (1993) Working Towards Rich & Flexible File Representations, in *Adjunct Proceedings of INTERCHI'93*, Amsterdam, The Netherlands. April 24-29.

Hill, W. C., Hollan, J. D., Wroblewski, D. A., & McCandless, T. P. (1991) "Edit Wear and Read Wear." In *Human Factors in Computing Systems: the Proceedings of CHI '91*. Austin,TX: ACM Press.

- Hough, M. (1990) *Out of Place*. New Haven: Yale University Press.
- Lakoff, G. & Johnson, M. (1980) *Metaphors We Live By*. The University of Chicago Press.
- Lynch, K. (1976) *Managing the Sense of a Region*. Cambridge: The MIT Press.
- Masinter, L., and E. Ostrom. (19xx) "Collaborative Information Retrieval: Gopher from MOO," Proceedings of INET '93.
- O'Sullivan, D. (1994). Yorb: An Electronic Neighborhood. Personal communication and unpublished videotape.
- Rose, D. "MUDs: (1994) Virtual Environments on the Net." Unpublished manuscript. Available from rose@apple.com).
- Southworth, M. (1969) The Sonic Environment of Cities. *Environment and Behavior*, June.
- Whyte, W. H. (1988) *City: Return to the Center*. New York: Doubleday.
- Wroblewski, D. A., McCandless, T. P. & Hill, W. C. (1994). Advertisements, Proxies, and Wear: Three Methods for Feedback in Interactive Systems. Natural Dialogue and Interactive Student Modeling[tentative title]. Heidelberg: Springer-Verlag, in preparation.